Demonstration of Sustainable Implementation-Level Workflow For Automating NFV Operation

Tatsuji Miyamoto, Masanori Miyazawa and Michiaki Hayashi
KDDI Research Inc.
2-1-15 Ohara, Fujimino-shi, Saitama, 356-8502, Japan
E-mail: {tt-miyamoto, ma-miyazawa, mc-hayashi}@kddi-research.jp

Abstract— **Maintainability of operation workflow is a critical issue for a telecom operator especially in a multi-vendor system environment. We demonstrate the implementation of an automated fault recovery workflow that is sustainable, versatile and applicable for fault management of both physical/virtual network functions.**

I. Introduction

Network function virtualization (NFV) is expected to reduce capital expenditure (CAPEX) and operational expenditure (OPEX) toward the realization of a future network that helps telecom operators enhance automated operation. In particular, the fault recovery operation has become a top priority because the recovery process in NFV is becoming increasingly complex in the case of adapting the existing manual operation due to the increase in managed resource components (e.g., server, operation system, hypervisor, and network functions). Thus, automated fault recovery (a.k.a. auto-healing) is expected in terms of improvement of service quality and reduction of OPEX by eliminating manual-based operation. In order to apply automated operation workflow in practice, the maintainability of the workflow is a significant indicator in view of reducing human error and the need for human effort. But, the modelling and implementation of the operation workflow are highly dependent on the developer's and vendor's specifications because there is no standard implementation-level workflow defining operational procedure. On top of that, the number of workflows, which are maintained and updated, increase because of further expansion of service, eventually leading to an increase in maintenance costs. To address the above problem, we proposed deriving a versatile implementation-level workflow for automated fault recovery in our companion paper presented at IM2017 [1]. The workflow is modeled by harmonizing ETSI NFV MANO into eTOM in order to break down the standard definition into detailed implementation-levels. To show the sustainability of the proposed workflow, we demonstrate fault recovery operation on a broadband service for most cases.

II. Testbed Configuration

Table I shows the system configurations in a demonstration. We implemented JBOSS BRMS 6 [2] as a workflow engine and the operation support system (OSS), which includes Zabbix 2.4 [3] as an alarm monitoring system (AMS). In order to automate the operation workflow, each task is implemented as an application programmable interface (API). For implementing MANO, we used our developed NFV Orchestrator (NFVO) called Othellon [4], OpenStack Tacker [5] as the VNF Manager (VNFM) and OpenStack Mitaka [6] as the virtual infrastructure manager (VIM).

Figure 1 shows the system architecture as an example for this demonstration. The network structure consists of both physical network functions (PNFs) and virtual network functions (VNFs) resources. For a demonstration of auto-healing by using the proposed workflow in [1], the testbed provides a broadband connection and cloud application services for a customer site with using a virtual CPE. The IP backbone network is constructed with physical routers. In data centers, a virtual CPE is deployed that consists of VyOS [7] as a virtual router and NAT, Opensense [8] as a virtual firewall and Dnsmasq [9] as a DHCP server. For deploying each node, physical computing resources are assigned as shown in Table I.

TABLE I. System configurations in Demonstration

	CPU	Mem size	Software spec
Server 1	Intel ® Xeon CPU @2.53GHz	16GB	Ubuntu14.04 (host OS)
WF Engine	2 core	8GB	JBOSS BRMS 6 [2]
AMS	1 core	1GB	Zabbix 2.4 [3]
Server 2 NFVO	Intel ® Xeon CPU @3.40GHz 8 core	32GB	Ubuntu12.04 (host OS) Othellon [4]
Server 3	Intel ® Xeon CPU @2.67GHz	8GB	Ubuntu14.04 (host OS)
VNFM, VIM	4 core	8GB	OpenStack Tacker [5] OpenStack Mitaka [6]
Server 4	Intel ® Xeon CPU @2.67GHz	8GB	Ubuntu14.04 (host OS)
vRouter	1 core	512MB	VyOS [7]
vNAT	1 core	512MB	VyOS
Server 5	Intel ® Xeon CPU @2.67GHz	8GB	Ubuntu14.04 (host OS)
vFirewall	1 core	1GB	Opensense [8]
vDHCP	1 core	512MB	Dnsmasq [9]
Router	-	-	-

978-3-901882-89-0 @2017 IFIP

III. Demonstration

For a demonstration, three scenarios of fault recovery operation were provided as shown in Table II. In order to recover a failure as rapidly as possible, generally telecom operators prepare one workflow for one pattern of failure in advance because the workflow becomes difficult to reuse because it is highly dependent on the developers' and vendor's specifications. However, the variety of workflows should be minimized to maintain operation automation. To address this problem, we proposed versatile workflow in [1], which is applicable for multiple patterns of failure, including both physical/virtual network functions. In the demonstration, the proposed workflow is implemented on JBOSS BRMS as shown in Fig. 2. The following describes the proposed workflow for each scenario.

Scenario 1 assumes a link down failure that is generated from a port of the physical router in Data Center A due to SFP module failure. When the OSS detects several alarms that include a "link down" failure of the physical router, firstly the OSS decides the healing pattern as PNF and executes a script of reset operations through the API. But the status of link down still remains in this case because of the hardware failure. Thus, the sequence of the workflow is forwarded to "Retry" and the decision task is executed again. The OSS decides the healing pattern as NS healing. Then, required resources are prepared on NFVI of Data Center B. After allocation is completed, NFVO requests VNFM to deploy virtual network function images of all CPE components which include DHCP, NAT, Router, and firewall. Finally, the OSS identifies the operation result and "clear" status, and the OSS finishes the workflow.

Scenario 2 assumes that a hardware server down has occurred that impacts the network connectivity of the virtual NAT and Router. When OSS detects alarms that include several "link downs" generated from a network port of the hardware server, virtual NAT and DHCP, OSS decides the healing pattern "VNF". Then, VIM allocates computer/network resources on the NFVI pool. When the allocation is completed, NFVO requests VNFM to deploy a virtual NAT and router on the allocated NFVI respectively.

Scenario 3 assumes a link down failure generated from a virtual DHCP and virtual firewall due to the occurrence of memory leak on NFVI. When OSS receives alarms of a virtual DHCP server and firewall, OSS decides the healing pattern "VNF" and requests for NFVO to allocate alternative resources for deploying the DHCP and firewall in series.

IV. Conclusions

We exhibited the proposed workflow in [1] with executing auto-healing of PNF, VNF and Network service (NS) to demonstrate the sustainability of the workflow. From the result, the workflow can deal with multiple fault scenarios, which include both PNF and VNF failure.

References

[1] T. Miyamoto, M. Miyazawa and M. Hayashi, "Sustainable Implementation-Level Workflow For Automating NFV Operation", in Integrated Network Management (IM), 2017 IFIP/IEEE International Symposium on. IEEE, 2017.

[2] JBOSS BRMS, http://www.jboss.org/quickstarts/brms/

[3] Zabbix, http://www.zabbix.com/

[4] K. Kuroki, M. Fukushima and M. Hayashi, "Framework of Network Service Orchestrator for Responsive Service Lifecycle Management", in Proc. IM 2015, pp. 960-965, 2015.

[5] OpenStack Tacker, http://docs.openstack.org/developer/tacker/

[6] OpenStack Mitaka, https://www.openstack.org/software/mitaka/

[7] VyOS, https://vyos.io/

[8] Opensense, https://opnsense.org/

[9] Dnsmasq, http://www.thekelleys.org.uk/dnsmasq/doc.htm

TABLE II. Scenarios of Demonstration.

Scenario	Failed resource	Healing pattern
1. SFP failure	Router	PNF →NS
2. Server down	vRouter, vNAT (Server 4)	VNF
3. Memory leak	vDHCP, vFW (Server 5)	VNF

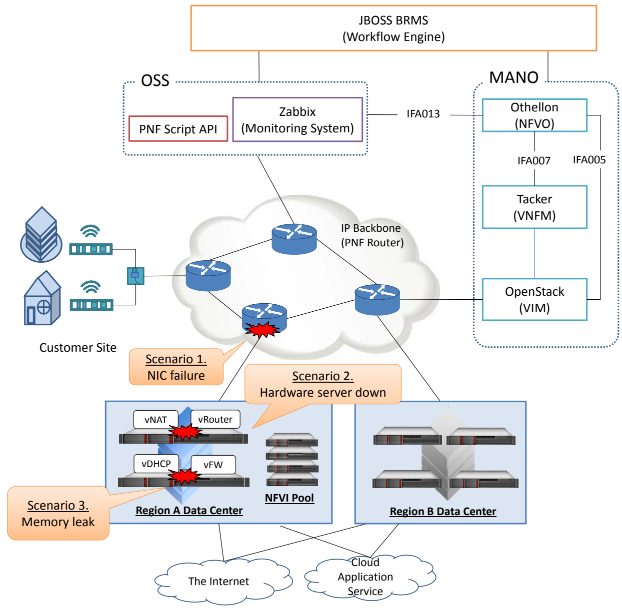

Fig. 1. System architecture of demonstration.

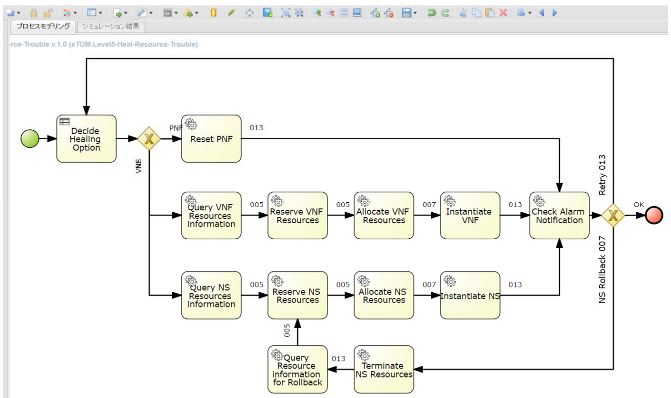

Fig. 2. Screenshot from the JBOSS BRMS indicates implementation of proposed eTOM Level 5 workflow "Heal Resource Trouble" in [1].